

T S2/5/1

2/5/1

DIALOG(R)File 347:JAPIO

(c) 2005 JPO & JAPIO. All rts. reserv.

07846582 **Image available**

PRINTER SYSTEM

PUB. NO.: 2003-341183 [JP 2003341183 A]
PUBLISHED: December 03, 2003 (20031203)
INVENTOR(s): SHIMADA NAOKI
APPLICANT(s): CANON INC
APPL. NO.: 2002-152365 [JP 2002152365]
FILED: May 27, 2002 (20020527)
INTL CLASS: B41J-029/38; G06F-003/12

ABSTRACT

PROBLEM TO BE SOLVED: To obtain a printer system utilizing the feature of an interface capable of hot-line insertion/draw-out in which print operation is sustained upon occurrence of a trouble during operation by connecting a new printer while operating a post PC or utilizing another printer connected already.

SOLUTION: The printer comprising the host PC and the interface capable of hot-line insertion/draw-out is further provided with a means for supporting the operational mode of the interface, a means for detecting a printer operable through a printer driver, and a means for delivering print data to a printer driver detected anew.

COPYRIGHT: (C)2004,JPO

?

T S4/3/1

4/3/1

DIALOG(R)File 345:Inpadoc/Fam.& Legal Stat

(c) 2005 EPO. All rts. reserv.

19566377

Basic Patent (No,Kind,Date): JP 2003341183 A2 20031203 <No. of Patents:
001>**PRINTER SYSTEM** (English)

Patent Assignee: CANON KK

Author (Inventor): SHIMADA NAOKI

IPC: *B41J-029/38; G06F-003/12

Language of Document: Japanese

Patent Family:

Patent No	Kind	Date	Applic No	Kind	Date
-----------	------	------	-----------	------	------

JP 2003341183	A2	20031203	JP 2002152365	A	20020527 (BASIC)
---------------	----	----------	---------------	---	------------------

Priority Data (No,Kind,Date):

JP 2002152365 A 20020527

?

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2003-341183
(P2003-341183A)

(43)公開日 平成15年12月3日(2003.12.3)

(51)Int.Cl. ⁷	識別記号	FI	キーワード(参考)
B 4 1 J 29/38		B 4 1 J 29/38	Z 2 C 0 6 1
G 0 6 F 3/12		G 0 6 F 3/12	D 5 B 0 2 1

審査請求 未請求 請求項の数5 OL (全13頁)

(21)出願番号 特願2002-152365(P2002-152365)

(22)出願日 平成14年5月27日(2002.5.27)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 島田 直樹

東京都大田区下丸子3丁目30番2号キヤノ

ン株式会社内

(74)代理人 100090538

弁理士 西山 恵三 (外1名)

Fターム(参考) 2C061 AP01 AQ05 HJ08 HQ02 HQ20

HV13 HV35 HV49

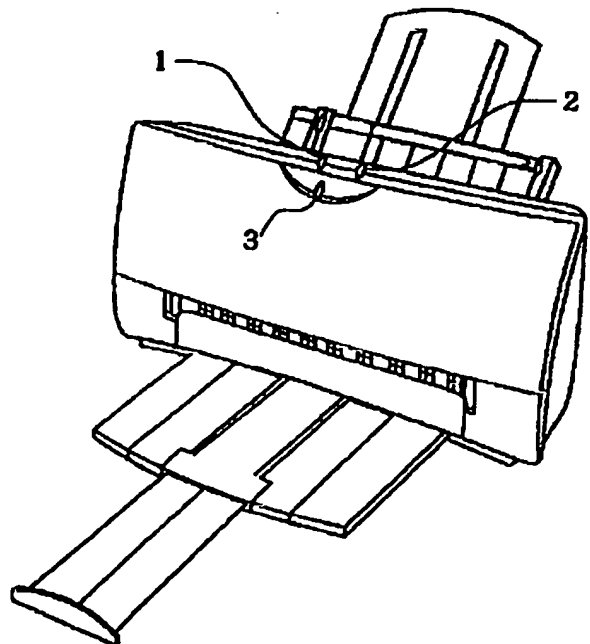
5B021 AA01 BB10 EE02 NN00

(54)【発明の名称】 プリンタシステム

(57)【要約】

【課題】 活線挿抜が可能なインターフェースの特徴を活用し、印刷中に障害が発生して印刷の継続が不可能となった場合、ホストPCを動作させた状態で、新しいプリンタを接続するか、もしくは既に接続されている別のプリンタを利用して、印刷を継続する。

【解決手段】 ホストPCと、活線挿抜が可能なインターフェースを備えたプリンタ。このプリンタにおいて、インターフェースの動作モードを支える手段。プリンタドライバによって、使用可能なプリンタを検出する手段。新しく検出されたプリンタドライバに、印刷データを送る手段。



【特許請求の範囲】

【請求項1】 活線挿抜 (Plug and Play) が可能なインターフェースを備えたプリンタ1において、第1のプリンタが印刷動作の継続が不可能となった場合、第2のプリンタを第1のプリンタの代わりに使用することを特徴とするプリンタシステム。

【請求項2】 前記請求項1のプリンタは、第1の動作モードと第2の動作モードを持ち、プリンタの第1の状態を検出する検出手段1と、同じくプリンタの第2の状態を検出する検出手段2と、プリンタのインターフェース動作モードを第1の動作モードから第2の動作モードへ変化させる変更手段1と、インターフェースの動作モードを第2の動作モードから第1の動作モードへ変更する変更手段2を備え、検出手段2が第2の状態を検出した場合には、変更手段1が前述のインターフェース動作モードを第1の動作モードから第2の動作モードへ変更し、検出手段1が第1の状態を検出した場合には、変更手段2が前述のインターフェースの動作モードを第2の動作モードから第1の動作モードへ変更させることを特徴とする、プリンタシステム。

【請求項3】 前記請求項2のプリンタシステムにおいて、第2の状態はプリンタ1が正常に印刷を行えない状態であることを特徴とするプリンタシステム。

【請求項4】 前記請求項2のプリンタシステムにおいて、第2の状態はプリンタ1が接続しているホストからデータを受信することができない状態であることを特徴とするプリンタシステム。

【請求項5】 前記請求項1のプリンタシステムにおいて、前記プリンタは動作状態をプリンタドライバへ通知する手段を有し、プリンタドライバは通知された動作状態を判断する手段を有し、前記判断手段が、前記プリンタが印刷を継続できないと判断した場合には、前記プリンタドライバが、印刷の継続が行える前記プリンタと同じホストに接続された別のプリンタを用いて印刷を継続させることを特徴とするプリンタシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、プリンタ等印字装置に関するものであり、特に一台以上のプリンタをホストコンピュータへ接続した場合の、プリンタ制御技術と印刷データ伝送技術の分野に関する。特に本発明では、接続されたプリンタが活線挿抜 (Plug and Play) が可能なインターフェースを備えているプリンタに関するものである。

【0002】

【従来の技術】 従来、プリンタが紙詰まり等の障害により印刷の継続が不可能となった場合、ユーザは印刷を中断して継続が不可能となった原因の解決を行わなければならなかった。

【0003】 ただし複数の高価なレーザービームプリン

タがネットワーク接続されている場合には、ネットワークを制御するサーバーの機能として、前述のサーバーが代替となりえる別のプリンタに対して、印刷データを送ることで連続した印刷を行うことが可能である。

【0004】 しかし安価なインクジェット方式のプリンタ等を考えた場合、その使用形態はネットワークに接続されている場合が希であり、むしろホストに対して直接接続されている使用形態がほとんどである。したがって、今現在もプリンタに障害が発生した場合には、その障害が解決される迄は印刷を再開することが不可能である。

【0005】

【発明が解決しようとする課題】 本発明は上記に記した、障害発生時に印刷が行えない点に注目して、その解決を行う物である。

【0006】 具体的には1台のホスト (=PC) に一台以上のプリンタを接続することを可能ならしめ、これにより、障害が発生したプリンタに送るデータを、別のプリンタに送ることで、前記障害が発生している間にも、印刷を継続させる。

【0007】 本発明に用いるプリンタはUSB (Universal Serial Bus) をインターフェースとして採用したプリンタである。USBは従来のセントロニクスインターフェースと比べて約10倍の転送速度を有する高速なシリアルバスである。またUSBの特徴としては、USB機器をカスケード接続することが可能である。さらにUSB機器は活線挿抜が可能なインターフェースであり、かつその接続や切断をホストPC側で検出することが可能である。

【0008】 現在、USBはホストPCに標準で搭載されており、特別なハードウェアを追加する必要は全く無い。

【0009】

【課題を解決するための手段】 本発明はかかる問題を解決するために、PCに接続したプリンタに障害が発生して、印刷の継続が不可能となった場合には、印刷データを別のプリンタに送る手段を提供するものである。

【0010】 (作用) これらの手段により、何らかの障害が発生した場合においても、印刷を継続することができ、従って最終的な印刷結果を得るまでの時間を短縮させることが可能である。

【0011】

【発明の実施の形態】 (実施例1) 以下に図面を参照しながら本発明の実施例について説明を行う。

【0012】 図1は本発明を実施したインクジェットによる印刷機構を有するプリンタの正面外観図である。1は電源スイッチ、2は紙無しエラー等の発生した障害から復帰するためのリセットスイッチである。3は電源投入時に緑色に点灯し、エラーが発生した時にはオレンジ色に点灯するLEDである。印刷データ受信時および印

刷時には点滅して、動作の違いをしらしめる。

【0013】図2はその背面図である。4はUSB (Universal Serial Bus) のAタイプコネクタの接続部であり5はBタイプコネクタの接続部である。6はセントロニクスタイプのコネクタである。尚、USBの規格に関しては、本発明の趣旨とは直接関係が無いので詳細な説明は省く。USBの規格にはUniversal Serial Bus Specification Version 1.0 (January 19, 1996) に詳しく書かれている。またセントロニクスタイプのインターフェース部はIEEE 1284 に準拠している。同様にIEEE 1284規格自体は、本発明とは直接的な関係が無いので、詳細な説明は省略する。

【0014】図3はUSBケーブルの外観図である。7はAタイプのコネクタであり8はBタイプのコネクタである。

【0015】図3のUSBケーブルを用いて図1のプリンタを図4のパーソナルコンピュータへ接続する。図4はパーソナルコンピュータの背面図である。図4の9はUSBのAタイプのコネクタが上下二段に実装されている例である。図3の7を図4の9の片方へ接続し、図3の8を図2の5へ接続する。同様に図2の4へ別なUSBケーブルのAタイプコネクタを接続し、同じケーブルのBタイプコネクタを、同じ形式の別のプリンタへ接続することも可能である。その様子を図5に示す。図5ではパーソナルコンピュータとプリンタとのUSB接続の様子を3例示している。図5の10はUSBケーブルである。100は図1のプリンタ、200は図4のPCを正面から見た図である。図5に示した3つの接続例のうち、接続例1は200のパーソナルコンピュータに100のプリンタを1台接続した例である。接続例2はパーソナルコンピュータの背面にあるUSBコネクタ(図4を参照)を使用して、100のプリンタを2台接続した例である。接続例3は4台のプリンタ100を1台のパーソナルコンピュータに、カスケード接続した例である。カスケード接続できるプリンタの台数は4台までである。カスケード接続はプリンタに実装されているUSBインターフェースがUSBハブの機能を持っているからである。USBハブは、USBの信号を分岐させる機能を有している。図15は55のUSB MPUを詳しく示した図である。55のUSB MPUは551のUSBハブと552のUSBインターフェースから構成されている。52(5)の入力は551のUSBハブで51(4)の出力と553の出力に分岐される。51(4)はカスケード接続を行うための外部出力で、553は55のUSB MPUへの出力である。552 USBインターフェースはUSBを経由して送られてきた信号を変換して、44のインターフェースコントローラへ送る。

【0016】この図1のプリンタは印刷用カートリッジを交換することが可能である。そのために、印刷する内容や目的に合わせた最適なカートリッジをプリンタに装着して印刷を行うことが可能である。図6は図1のプリンタに装着可能であるところの印刷カートリッジ各種である。

【0017】図6の11は黒インク単色の印刷用カートリッジである。テキストだけのドキュメント等を印刷する場合に最適なカートリッジである。図6の12は写真画像印刷用のインクカートリッジである。デジタルカメラなどで撮影した画像の印刷には最適なカラーインクカートリッジである。図6の13は通常のカラー印刷用のカートリッジである。一般的なカラー印刷に用いると、最適な結果を得ることが出来る物である。本発明を実施した図1のプリンタは、これらのカートリッジを装着することが可能である。その方法は図7に示されている。

【0018】図7の14はカートリッジ交換スイッチである。このスイッチを押すことにより、カートリッジ300がキャリッジ400の中央へ移動する。これはカートリッジの交換を容易に行うためである。図7の15はカートリッジを取り外す場合を示している図である。図7の16はカートリッジを装着している図であり、図7の17はカートリッジを装着する状態を拡大して図示したものである。

【0019】図8は本発明を実施した図1のプリンタのブロック図である。図8の18は図1の1に相当する電源スイッチである。図8の19は図7の14に相当するカートリッジ交換スイッチである。図8の20は図1の2に相当するリセットスイッチである。図8の21は電源投入時に緑色に点灯するLEDであり、図1の3に相当する。図8の22はエラー発生時にオレンジ色に点灯するLEDで、同じく図1の3に相当する。このLEDは単体で緑色とオレンジ色の2色に点灯することが可能である。図8の23は図8の18電源スイッチの操作を図8の30であるMPUへ伝達する信号線である。この信号は図8の26の割込み信号として図8の30であるMPUへ伝達されるため、30のMPUにおいて、優先的に処理される。図8の25は図8の27であるプリンタ制御ゲートアレー、30のMPU、29のROMを接続するアドレスバスとデータバスである。29のROMはMPUの動作、処理を記述したプログラムが記録されている。図8の27はプリンタ制御ゲートアレーであり、スイッチ、LED、インターフェース、メモリなどの制御機能を有する。図8の28は図8の30であるMPUが処理を行う場合にデータ等を一時的に記憶する記憶領域を提供するRAMである。ここのRAMの一部は受信した印刷データを一時的に蓄えておくバッファとしても使用される。図8の31は図8の32及び33であるステップモータの制御を行う信号を伝達する信号線である。32はインクカートリッジの支持体である図7の

103であるキャリッジを、図1正面方向から見て左右に移動させるキャリッジモータである。33は印刷用紙をフィードするフィードモータである。印刷用紙は図1の100である印刷用紙トレイに収容され、プリンタ内部を通過して101の印刷用紙受けに排出される。この印刷用紙の移動を行うのが33のフィードモータである。図8の35は図6に示されているインクカートリッジである。このインクカートリッジは、インク自体とそれを吐出するヘッドが一体となった構造である。またインクカートリッジ自体の種別を示すIDも記録している。図8の34は、35のインクカートリッジにあるヘッドの制御及びIDの読み出しを行う制御線である。

【0020】図8の36は一般の家庭に提供されている商用電源である。図8の37はこの商用電源を図8の38である直流5ボルトと39の直流24ボルトへ変換するACアダプターである。38は30のMPU等の電子部品が動作するために使用され、24ボルトは32などのモータ及び35のインクカートリッジにあるヘッドを駆動するために使用される。

【0021】図8の40は44のインターフェースコントローラと25のプリンタ制御ゲートアレーとを接続するインターフェースコントロール信号である。これはインターフェースの動作モードを27プリンタ制御ゲートアレーが検出するために用いられる。41はインターフェースゲートアレーである44がPCから受信したデータを27へ送るデータバスである。この44インターフェースコントローラは、46のセントロニクスコネクタからのデータと55のUSB MPUからのデータの切り替えを行う物である。この切り替えは42のアナログスイッチで行う。44インターフェースコントローラは常に55USB MPUからのデータを優先する様に設計されている。即ちセントロニクスインターフェースである46と51のUSB Bタイプコネクタの両方に印刷データが送られてきた場合には、常にUSB側を優先し、セントロニクス側からのデータは受信しない。この切り替えは、43のセントロニクスインターフェースからの信号線と47の信号線のステータスを用いて行われる。45はセントロニクスインターフェースからのデータ信号である。48は55のUSBインターフェースを制御するUSB MPUのプログラムが記憶されているフラッシュメモリである。既に述べたようにUSBにはAタイプとBタイプのコネクタがあり、PCもしくは上流の機器には52Bタイプコネクタが接続される。また下流の機器には51Aタイプのコネクタから接続される。53と54はそれぞれ51のAタイプコネクタと52のBタイプコネクタへの信号線である。

【0022】次に本発明を実施したプリンタドライバを実行するオペレーティングシステム（以下OSと表記する）マイクロソフト社のWindows（登録商標）95の印刷メカニズムに付いて簡単に説明を行う。図9の

56はアプリケーションであり、ユーザはこの56アプリケーションでドキュメントの編集等の作業を行う。ユーザは56のアプリケーションに対して、作成したドキュメントの印刷指示を行う。印刷の指示は印刷を行うプリンタをアプリケーションから指定して行う。指示をする画面の様子を図10に示す。使用するプリンタ名称が図10の77に示されている。

【0023】プリンタを指定することで、OSの描画プロセスであるGDI（図9の57）に対して、プリンタの特性に関する情報を渡す。具体的にはカラー印刷が可能であるか、印刷に用いる用紙のサイズなどである。57のGDIはこれらの情報を元にして、ドキュメントの描画を行う。57のGDIが描画を行なった結果は、OS固有のフォーマットであり、このフォーマットは印刷のみならずCRTモニタ等への表示のためにも用いることが可能である。

【0024】この57GDIの出力結果を59のプリンタドライバへ入力する。このプリンタドライバは、使用するプリンタの機種それぞれに固有であり、通常プリンタメーカーが提供するものである。この59プリンタドライバは57GDIからの出力を、プリンタが使用するインクの特性に最適となるように変換を加え、さらにそれらをプリンタへ送り出す印刷データを作成する。

【0025】59プリンタドライバが作成した印刷データは62のGDI32へ渡される。62のGDI32はスプーラプロセスと59のプリンタドライバのインターフェース的なモジュールである。これは59のプリンタドライバが16bitアプリケーションであり、63のスプーラプロセスが32bitアプリケーションであるためである。通常16bitアプリケーションは32bitアプリケーションと直接呼び出すことができない。そのため16bitアプリケーション用から呼び出される部分と、32bitアプリケーションから呼び出される部分の両方を持つ62のGDI32が必要となる。

【0026】62のGDI32を経由して63のスプーラプロセスを送られた印刷データは65のスプーラヘービ記録される。印刷データのスプールが終了した後、63のスプールプロセスは65のスプーラから印刷データを読み出して、それを69のLanguageモニタへ送る。この69Languageモニタは、プリンタのステータス、例えば印刷用紙が無くなった場合等の原因で印刷データが受信できない状態等、を検出して必要に応じてユーザに対して通知したり、印刷データのプリンタへの送出を開始・停止を制御するゲートの役割も持つ。そのために、印刷データの内容を解析する機能も有している。この機能には、印刷データに含まれる制御コマンドを検出し、ページの境界を検出する機能も含まれる。図11から図12は2ページのドキュメントを56のアプリケーションで作成し、その印刷を行った場合における、59が生成した印刷データである。図11の

80、0CHは紙送りコマンドである。76のプリンタはこのコマンドを受信すると、印刷を行っていた用紙を排出し、80に継続するデータがある場合には、新しい印刷用紙をフィードする。

【0027】本発明を実施したプリンタはUSBをインターフェースとしているため、69のLanguage モニタはPCのUSBポートへ印刷データを送り出す。ポート自体はハードウェアであるため、ポートを直接制御するモジュールは、プリンタドライバが実行されるRing 3では無く、特権モードといわれるRing 0で実行される。Ring 3側のアプリやモジュールはRing 0側のモジュールに直接アクセスすることはできないので、ハードウェアであるところの75 USBポートをRing 3にあるモジュールに対して仮想化しているのが70のUSBポートモニタである。Ring 3側のモジュールは、この70のUSBポートモニタへアクセスすることで、直接ハードウェアをアクセスしているのと同様の結果を得ることができる。70のUSBポートモニタは71のOSが提供するWIN32SPIへアクセスする。このモジュールはRing 3とRing 0の橋渡しをするモジュールである。

【0028】USBのハードウェアである75のデバイスドライバはOSベンダから提供される。74のUSB 共通ドライバは、USBのハードウェアを直接制御するモジュールである。74 USB 共通ドライバの提供する機能を統合して、プリンタ向けの機能を提供しているのが、72のUSBプリンタドライバである。72のUSBプリンタドライバは、71のWIN31API経由で送られてきた印刷データを74のUSB 共通ドライバへ送り出すことで、最終的には76のプリンタへ印刷データが送られて印刷が行われる。

【0029】図13はWindows（登録商標）95におけるプリンタ用ポートの情報がレジストリに記録されている情報の例を示している。プリンタ用USBのポートはUSBPRN01からUSBPRN10までの10ポート用意されている。これらUSBPRN01等のポート名称は、図5のようにカスケード接続されたプリンタを区別するために、便宜的に付けた名称であり、実際のUSBインターフェースは一つだけである。OSにおけるプリンタの管理方法は、ポート毎にプリンタが一台接続されると言う方法であるため、印刷データの送り先としてOSレベルでプリンタを識別する方法は、ポート名称を用いて行う。

【0030】これらのポートの割り当て方法について説明を行う。パーソナルコンピュータに図1のプリンタが1台だけ接続されている場合には、このプリンタにはUSBPRN01のポートが割り当てられる。さらにもう1台接続された場合にはUSBPRN02が割り当てられる。同様に接続するプリンタの台数が増えるにつれて、ポート番号は順次増加する。このように割り当てら

れるポート番号は、特定の時点でもっとも古く接続されたプリンタがもっとも小さいポート番号を占有する。

【0031】またUSBPRN01のポートを割り当てられたプリンタのパーソナルコンピュータとの接続をやめた場合、USBPRN01は消滅する。しかし次にプリンタが接続された場合には、USBPRN01が復活して使用可能となる。

【0032】なお、既に説明したように、同時に使用できるプリンタは最大4台までである。

【0033】図14はUSBPRN07に接続しているプリンタに関する情報である。この情報も図13と同じくWindows（登録商標）95のレジストリに記録されている。141は接続しているポート名称である。また142は使用しているプリンタドライバの名称である。143は登録されているプリンタの名称を示している。

【0034】次にプリンタ1台を接続して印刷を行っている状態で、そのプリンタに障害が発生した場合における本発明の処理例について説明を行う。これは図5の接続例1の場合である。

【0035】始めにプリンタ側の処理について、図16のフローチャートを用いて説明する。

【0036】次に69のLanguage Monitorと70のUSB Port Monitorにおける処理を図17のフローチャートを用いて説明を行う。なお、本発明とは直接的な関係が無い部分については、簡略して説明を行う。

【0037】図1の1電源スイッチを押下されてプリンタの電源がOnになると、まずS1でプリンタ全体の初期化処理を行う。次にS2で図1のプリンタのステータスチェックを行う。S3でプリンタをOFFにする要求があるかのチェックを行う。この要求は図1の1である電源スイッチを再度押下した場合に発生する。プリンタOFF要求がある場合には、S7において終了処理を行う。S7の終了処理には、プリンタの37であるACアダプタからの24V出力39をオフする処理も含まれている。S4では何らかの印刷継続の障害となるようなことが発生しているかどうかの検出を行う。発生していない場合にはS5で印刷データを受信し、S6で印刷処理を行う。障害が発生している場合には、S8において552のUSBインターフェースが使用可能な状態であるかを調べる。既に使用不可となっている場合にはS2へ戻る。使用可能な状態の場合には、S9において、552のUSBインターフェースを使用不可とする。この状態は再度初期化処理がS1で行われるまで解除されない。

【0038】S8においてUSBインターフェースが使用不可の設定となった場合は、図15において551のUSBハブだけが動作している状態である。従ってUSBプリンタとしての機能はホストであるPCからは認識

されなくなる。しかし551のUSBハブは動作しているので、仮にこのプリンタに、さらに別のプリンタが接続されている場合には、その接続と機能は維持される。

【0039】続いてパーソナルコンピュータ側の70 USBポートモニターで行われる処理について、図16のフローチャートを用いて説明を行う。

【0040】印刷データをポートへ書き込む前に、S10で書き込むポートをOpenする。この場合ポートはUSBPRN01である。次にS11にて70のUSBポートモニタの上位モジュールである69Language Monitorより印刷データを受信する。S12において印刷データが終了したかをチェックし、終了している場合にはS15でOpenしたポートをCloseする。

【0041】印刷データが終了していない場合には、S13でポートにデータを書き込む。続いてS14において、書き込みが成功したかどうかをチェックし、成功した場合にはS11へ戻り次の印刷データを受信する。

【0042】プリンタに障害が発生した場合には、プリンタ側にてUSBインターフェースが使用不可能とするため、今まで書き込んでいたポートが消滅してしまうことになる。そのためS14では書き込みが失敗したと判定される。そこでS17にて、印刷データの書き込みが行えるPortの検出を行う。既に説明した通り、USBは活線挿抜が可能なインターフェースであるので、この時点で別なプリンタを接続することが可能である。また、従来から接続されているプリンタの障害が解消して、再度印刷が可能となる場合もある。いずれの場合も動作が可能となるプリンタが現れた場合は、最初のポートであるUSBPRN01のポートが復活することになる。このようなポートの状態の変化が発生すると、S17で検出される。

【0043】S17でポートが存在すると判断されると、S19でその先に接続されているプリンタの検出を行う。このとき先程まで印刷に使用されていたプリンタと同一のプリンタである場合、すなわち障害が解決して再び動作が可能となった場合は、S21にてプリンタ内部のバッファ28にデータが残っているかどうかをチェックする。残っている場合にはデータの連続性が確保されているので、S13で引き続き印刷データをポートに書き込む。

【0044】同じプリンタでは無い場合にはS22で同じ印刷機能を提供できるプリンタかどうかをチェックする。既に説明したとおり、本発明を実施したプリンタは、3種類のインクカートリッジを交換することが可能である。59のプリンタドライバは、このカートリッジの種類に最適となるように印刷データを作成する。従って仮に新しく接続されたプリンタが同じ種類であったとしても、装着されているインクカートリッジの種類が違えば、既に作成された印刷データを用いても、期

待された印刷結果を得られない。たとえば最初に用いていたプリンタに装着されていたインクカートリッジが13のカラー印刷用であった場合、59のプリンタドライバが作成した印刷データもこの13に最適なカラー印刷用データである。しかし新しく接続されたプリンタが11の黒インク単色のカートリッジが装着されている場合には、期待したカラー印刷が行えない。そのためにS22で同じ印刷機能を提供することが可能なプリンタかどうかをチェックする。提供できないと判定された場合には、S23でポート番号を増やして、すなわちUSBPRN02としてポートの検出をS17で行うようにする。ここではまた新しいプリンタが接続されていないので、S18でポート無しと判断される。S17におけるポートの検出について、たとえ検出するポートの番号がUSBPRN02であったとしても、再び接続されているプリンタの数が減った場合、すなわちポートの数が減った場合には、再びUSBPRN01からチェックを始める。

【0045】S22で同じ機能を提供できるプリンタが接続されたと判断された場合には、S24にて書き込み途中の印刷データは廃棄する作業を69のLanguage Monitorに依頼する。69のLanguage Monitorは印刷データの内容を認識できるので、次ページの印刷データの開始まで印刷データを読み飛ばす。具体的に印刷用紙送りを示すOCHまでを読み飛ばす。最初に接続していたプリンタ内部のバッファ28に、ある程度の印刷データが蓄えられてしまっているため、新しく接続されたプリンタには印刷データの途中から送り出すことになってしまう。その場合、印刷データの途中から受信したプリンタでは、それらを正しく解釈できないため不適切な印刷を行ってしまう。これを防ぐ目的で、現在印刷しているページの印刷データを廃棄する。印刷データの廃棄が終了した後は、次ページのデータを書き込むためにS11にて69のLanguage Monitorから印刷データを受信する。

【0046】(実施例2) 実施例1では、印刷が継続できない状態が発生した場合には、プリンタ側においてインターフェースを使用不可とし、障害発生後に接続されたプリンタもしくは既に接続されているプリンタを用いて印刷を継続する方法を示した。

【0047】実施例2では同様の機能をドライバで実現する例について説明を行う。

【0048】そのためには図18に示したように、図9の印刷メカニズムに対して、いくつかのモジュールを追加する。

【0049】59のプリンタドライバは作成した印刷データを120の第1のスーパーへ送る。121のバックグラウンドは120の第1のスーパーから印刷データを読み出して、122のマルチポートコントローラへ送る。このマルチポートコントローラは印刷データをページ単

位で分割する作業を行う。分割された印刷データは、123のUSB出力タスクへ送られる。123のUSB出力タスクは、分割された印刷データとページ順序を関連づけて、ページ単位で管理を行う。

【0050】通常の印刷であるならば、123のUSB出力タスクは1ページ目から順番に印刷データを63のスーパープロセスを送り出す。

【0051】既に説明した通り、USBをインターフェースとする図1のプリンタは図5に示すとおり複数台数1台のプリンタに接続することが可能である。従って、63のスーパープロセスは同時に接続されているプリンタの台数分作成される。例えば4台のプリンタがそれぞれUSBPRN01、USBPRN02、USBPRN03、USBPRN04へ接続されている場合には、スーパープロセスもポート毎に4つ作成される。

【0052】124のステータスチェックは接続されたプリンタの状態をチェックするモジュールである。接続されているプリンタに、障害が発生して印刷が継続できない状態となった場合には、この124ステータスチェックが認識する。124のステータスチェックが認識した状態に基づいて、122のマルチポートコントローラは印刷データの配分先を決定する。

【0053】その他の印刷メカニズムに関する点は、既に実施例1において図9にて説明したものと同様である。

【0054】続いて複数のプリンタがホストであるパーソナルコンピュータに接続されている状態で、そのうちの1台に障害が発生して印刷が継続できなくなった場合の説明を行う。説明には図18のフローチャートを使用する。

【0055】例として図5の接続例3に示した4台のプリンタがパーソナルコンピュータに接続されている状態を用いて説明する。接続されたプリンタはプリンタ1からプリンタ4として、それぞれにUSBPRN01からUSBPRN04までのポートが割り当てられているものとする。説明は図19のフローチャートを使用する。

【0056】プリンタ1向けの印刷データの場合、S20で121のバックグラウンドからの印刷データの到着をチェックする。到着するとS21でバックグラウンドより印刷データの読み出しを行う。S22で印刷データの終了をチェックして、終了の場合には、処理を終了する。次にS23で読み出した印刷データのページエンドをチェックする。ページエンドが見つかった場合には、S24で123にて出力タスクを生成する。この時点で1ページ分の印刷データができあがったので、S25にて、そのデータを順次63のスーパープロセスへ出力する。S26で123のステータスチェックにアクセスして、スーパープロセスに送り出した印刷データが、全てプリンタ1に送り出されたかをチェックする。全ての印刷データがプリンタ1に送り出され、プリンタ1において印刷も成功した場合には、S27作成したタスクを破棄する。

【0057】プリンタ1用の印刷データがプリンタ1に送り出しに失敗した場合には、S28が124のステータスチェックにアクセスして、他に使用可能なプリンタの存在をチェックする。もし他に使用可能なプリンタが存在しない場合には、S29で印刷を継続するかどうかのメッセージを表示する。S30で操作者の指示を判断し、印刷終了の指示をした場合には、処理を終了する。

【0058】この時点でUSBPRN02、USBPRN03、USBPRN04のポートとそれぞれに接続しているプリンタ2、プリンタ3、プリンタ4があるので、S24ではこれらのプリンタが見つかる。最初に見つかるのはUSBPRN02である。S31でプリンタ2が使用中であるかどうかをチェックし、使用中でない場合にはプリンタ1の代わりにプリンタ2を使用するために、S32で先程プリンタ1に送出できなかったタスクをプリンタ2へ送り出す。そしてS27でタスクを破棄する。

【0059】プリンタ2が使用中である場合には、プリンタ3を使用する。

【0060】全てのプリンタが使用中の場合には、新しいプリンタが接続されるか、印刷を終了させる。新しいプリンタが接続された場合には、そのプリンタにはUSBPRN01が割り当てられ、プリンタ1として使用される。

【0061】USBインターフェースは他のプリンタが使用中でも、接続されたポートを新規に登録することが可能である。

【0062】

【発明の効果】以上説明した通り、活線挿抜が可能なインターフェースを用いたインターフェースを、プリンタに採用することにより、印刷を継続が不可能な状態（印刷用紙切れ、インクカートリッジのインク切れ等の障害）となっても、印刷を中断することが無い、もしくは短時間で印刷を再開することが可能となる。

【0063】その結果、障害を取り除いている間にも、時間を無駄にすること無く印刷を行えることが可能となる。

【図面の簡単な説明】

【図1】本発明を実施したプリンタの正面図。

【図2】プリンタの背面図。

【図3】プリンタのインターフェースであるUSBのケーブルの外観図。

【図4】PCの背面にあるUSBコネクタの図。

【図5】USBプリンタの接続例を示す図。

【図6】プリンタに装着可能なカートリッジの外観図。

【図7】カートリッジの脱着方法を示した図。

【図8】プリンタの電気回路ブロック図。

【図9】実施例1のプリンタドライバのブロック図。

【図10】アプリケーションから前述のプリンタドライバを用いて印刷を行う際に表示されるダイアログの図。

【図11】印刷データの例。

【図12】印刷データの例。

【図13】レジストリに記録されたプリンタポートの例。

【図14】レジストリに記録されたプリンタの情報の例。

例。

【図15】USB MPUを説明したブロック図。

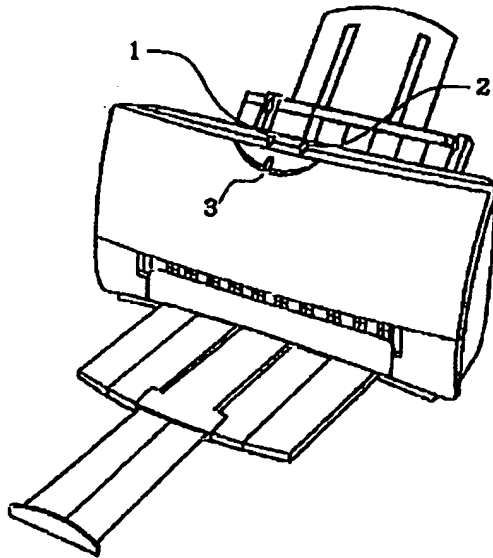
【図16】実施例1におけるプリンタ側の処理を説明したフローチャート。

【図17】実施例1におけるポートモニタにおける処理を説明したフローチャート。

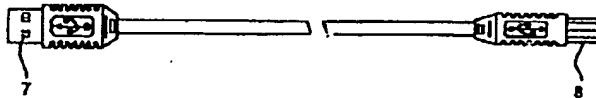
【図18】実施例2のプリンタドライバのブロック図。

【図19】実施例2の処理を説明したフローチャート。

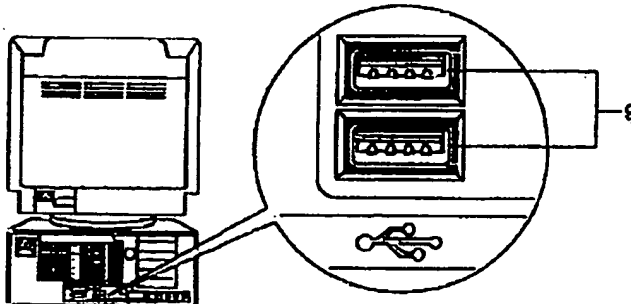
【図1】



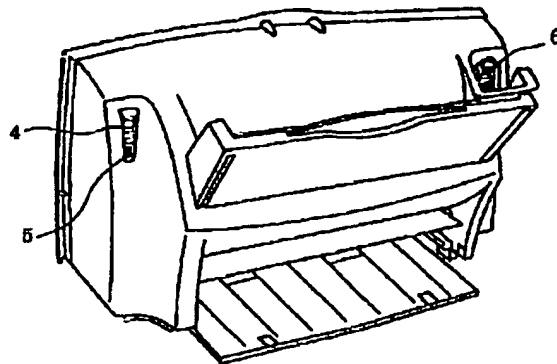
【図3】



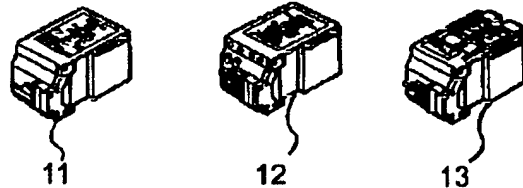
【図4】



【図2】



【図6】

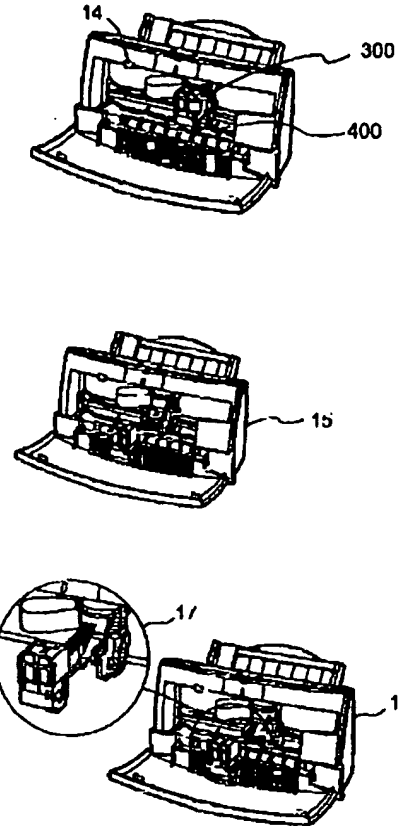


【図12】

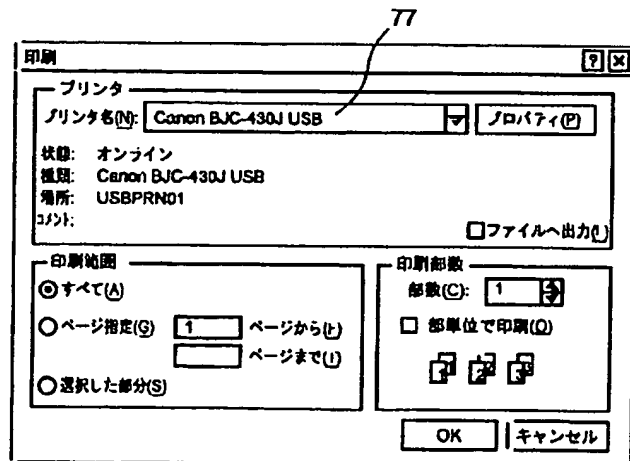
480: 65020000 01182841 08004802 00030720	a.... (A..Kj....
490: 0700001B 28650200 00011828 41080048	.. (a.... (A..K
4A0: D2000303 00078000 1B286502 0000011B	.. (a.... (A..K
4B0: 28410600 48D00001 0F80081B 28650200	(A..Kj.... (e....
4C0: 00011828 41050048 D000001F 001B2865 (A..Kj.... (e
4D0: 02000001 1B284105 00480000 003E0D1B (A..Kj.... <
4E0: 28650200 00011828 41050048 D000003C	(e.... (A..Kj.... <
4F0: 0D1B2865 02000001 1B284105 00480000	(a.... (A..Kj....
500: 0078001B 28650200 00011828 41050048	x.... (a.... (A..K
510: D00000F0 0D1B2865 02000001 1B284105	.. (a.... (A..K
520: 0048D100 0101E000 1B286502 0000011B	.. (a.... (A..K
530: 28410600 48D10001 03C0001B 28650200	(A..Kj.... (e....
540: 00011828 41060048 D1000107 80001B28 (A..Kj.... (
550: 65020000 01182841 050048D1 00000F0D	e.... (A..Kj....
560: 1B286502 0000011B 28410500 48D10000	(a.... (A..Kj....
570: 0C0D1B28 65020000 01182841 070048D1 (a.... (A..Kj....
580: 00021800 60001B28 65020000 01182841 (a.... (A..Kj....
590: 070048D1 00023000 60001B28 65020000	.. (A..Kj.... (
5A0: 01182841 070048D1 00023000 E0001B28 (A..Kj.... (
5B0: 65020000 01182841 070048D1 0002C000	e.... (A..Kj....
5C0: E0001B28 65020000 01182841 080048D2	.. (a.... (A..Kj....
5D0: 00030180 00E0001B 28650200 00011828 (a.... (A..Kj....
5E0: 41080048 D2000303 E001E000 1B286502	A..Kj.... (a....
5F0: 0000011B 28410600 48D20000 07FFF700 (A..Kj....
600: C00D1B28 65020000 01182841 090048D2	.. (a.... (A..Kj....
610: 000007FF FF00C000 1B286502 0000011B (a.... (A..Kj....
620: 28410600 48D20000 0FFFFF00 C00D1B28	(A..Kj.... (a....
630: 65020000 01182841 090048D2 00001FFF (A..Kj....
640: FF00C000 1B286502 0000011B 28410600	.. (a.... (A..Kj....
650: 48D20000 1FFFFF00 C00D1B28 65020001	Kj.... (a....
660: 790C1B28 62010000 1B286501 00001B40	y.... (a....

BEST AVAILABLE COPY

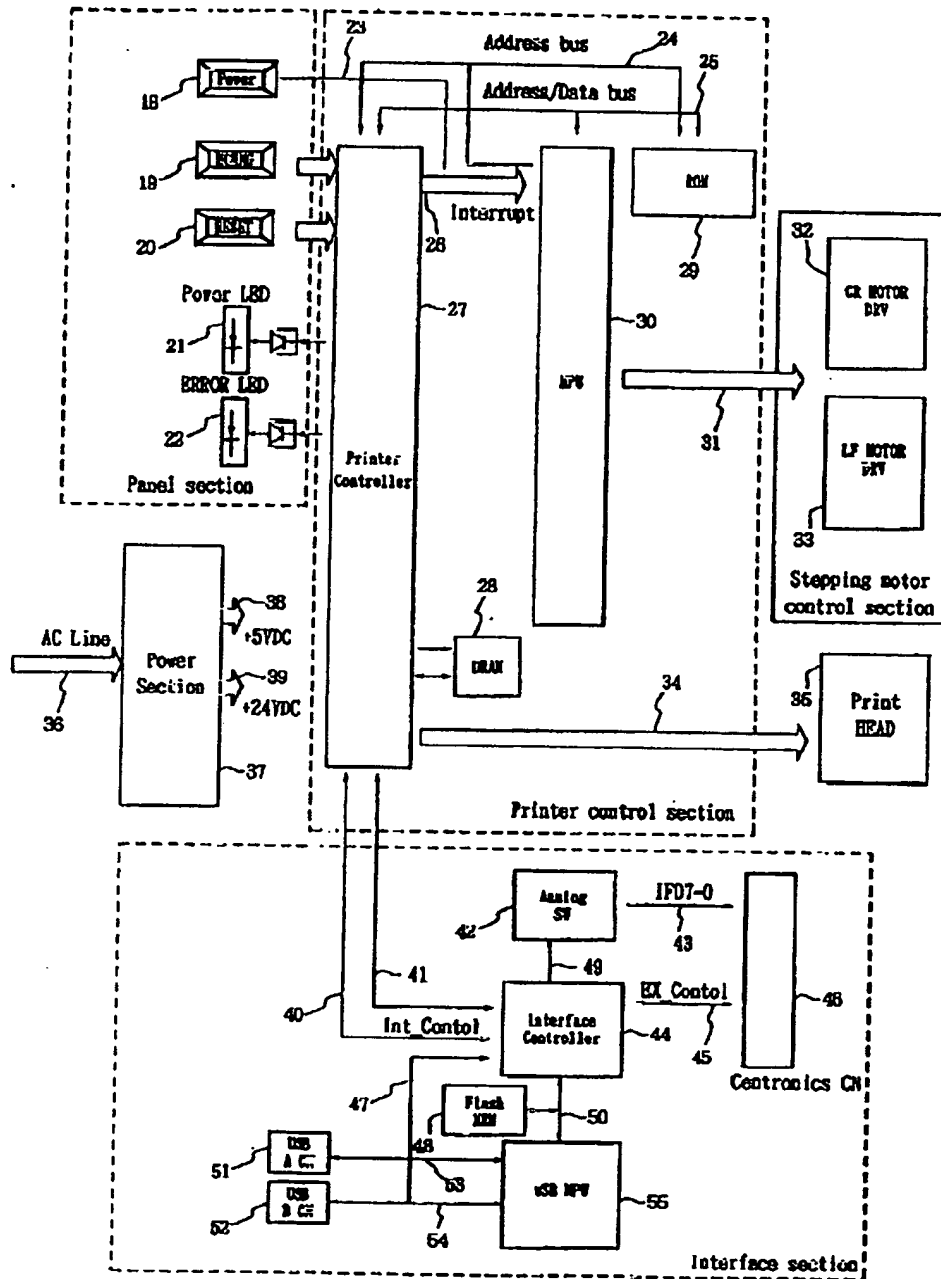
【図7】



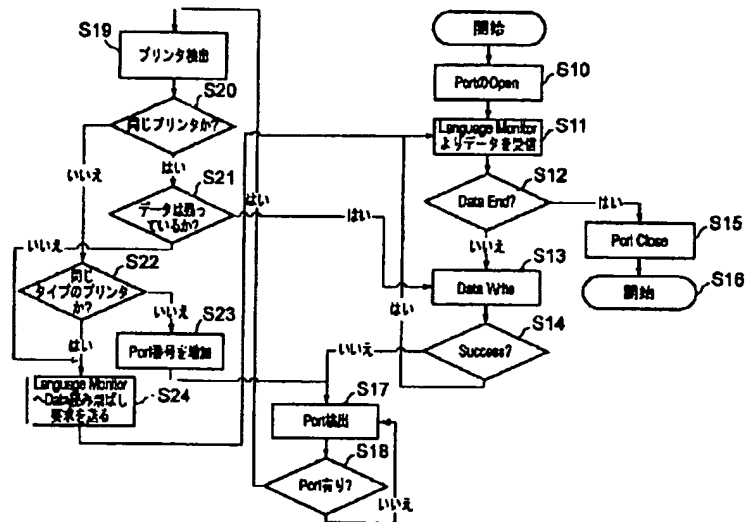
【図10】



【図8】



【図17】



【図18】

